

# Planeamiento de condensadores en sistemas de distribución radial usando entrenamiento neuronal

## Planning of Condensers on Radial Distribution Sistem Using Neuronal Training

**MAURICIO GRANADA ECHEVERRI**

Ingeniero eléctrico y magíster en Ingeniería eléctrica en la Línea de planeamiento eléctrico de la Universidad Tecnológica de Pereira. Docente de la Facultad de Ingeniería Eléctrica de la misma universidad.

[magra@utp.edu.co](mailto:magra@utp.edu.co)

**CARLOS ALBERTO RÍOS PORRAS**

Ingeniero eléctrico y magíster en Ingeniería eléctrica de la Universidad Tecnológica de Pereira. Profesor del programa de Tecnología eléctrica de la misma Universidad.

[alpor@utp.edu.co](mailto:alpor@utp.edu.co)

**RAMÓN ALFONSO GALLEGO RENDÓN**

Ingeniero eléctrico de la Universidad Tecnológica de Pereira, magíster en Ingeniería eléctrica de la Universidad Nacional de Colombia y doctor en Ingeniería eléctrica de la Universidad de Campiñas (Brasil). Docente de la Facultad de Ingeniería eléctrica de la Universidad Tecnológica de Pereira.

[ragr@utp.edu.co](mailto:ragr@utp.edu.co)

Clasificación del artículo: investigación

Fecha de recepción: 18 de noviembre de 2005

Fecha de aceptación: 2 de junio de 2006

**Palabras clave:** redes neuronales, condensadores, optimización combinatoria, redes de distribución.

**Key words:** neuronal networks, condensers banks, combinatorial optimization, distribution networks.

### RESUMEN

En este artículo se utiliza una red neuronal multicapa tipo *Back Propagation* (BP) para resolver el problema de la localización y tamaño óptimo de bancos de condensadores en sistemas eléctricos de distribución, con el fin de reducir las pérdidas del

sistema en un horizonte de tiempo determinado y considerando que el problema es de programación no lineal entero mixto (PNLEM) con alta explosión combinatoria. Para calcular el conjunto de validación de la red neuronal se emplean los resultados de investigaciones anteriores, obtenidos usando la meta-heurística denominada *búsqueda tabú*. Se

pretende verificar las características de generalización de la red neuronal en problemas PNLEM y agilizar el proceso de optimización, para obtener resultados de buena calidad en tiempo real, dado que el esfuerzo computacional se disminuye significativamente.

#### ABSTRACT

In this article a multilayer neuronal network with Back Propagation (BP) training is used to solve the

optimal location and size problem of condensers banks in distribution electrical systems to reduce the system losses in a time horizon, considering this is a Mixed Integer Nonlinear Programming (MINP) problem with high combinatorial explosion. To calculate the neuronal network validation set, the results obtained from previous investigations using *tabu search* were used, to verify the neuronal network generalization characteristics and accelerate the optimization process.

\* \* \*

## 1. Introducción

En un sistema de distribución de energía eléctrica se emplean bancos de condensadores para compensar la potencia reactiva de las cargas; esto trae una serie de beneficios como la disminución de pérdidas de energía, el mejoramiento del perfil de tensión y el incremento de la capacidad del sistema, entre otros. La compensación de la potencia reactiva está directamente relacionada con la ubicación, el tamaño, el número y el costo de los bancos de condensadores.

Dado que las cargas en los sistemas de distribución se caracterizan por tener consumos de energía dinámicos o variables, existen infinitas posibilidades para realizar la compensación. Este problema ha sido modelado matemáticamente como de programación no lineal entero mixto (PNLEM), y en su solución se busca localizar y dimensionar en forma óptima los bancos de condensadores para reducir los costos de operación, las pérdidas y la inversión en la instalación. En la literatura especializada su solución ha sido abordada a través de métodos como: la optimización por gradiente, la programación dinámica, los algoritmos genéticos [1], y las técnicas *simulated annealing* y *tabu search* [2], entre otros.

Los métodos de solución mencionados anteriormente tienen la dificultad de presentar un elevado esfuerzo computacional, lo que los hace poco atractivos

para ser implementados en sistemas de control automatizado en tiempo real. Por su parte, las redes neuronales ofrecen las ventajas de «aprender» un conjunto de patrones previamente entrenado y generalizar sus respuestas frente a nuevos conjuntos con un tiempo de reacción casi inmediato.

Partiendo de esta idea resulta atrayente entrenar una red neuronal con las configuraciones óptimas de los condensadores para unas condiciones de carga específicas en un sistema de distribución y, luego, utilizar la capacidad de generalización de la red neuronal para reconocer cualquier condición de carga y definir la configuración óptima de condensadores. En este trabajo, las configuraciones óptimas de condensadores fueron obtenidas usando el algoritmo optimizador *tabu search*. Además, la codificación usada para ingresar la información a la red marca una importante diferencia en el desempeño de la misma.

Actualmente, las acciones de control de los bancos de condensadores son ejecutadas por mecanismos individuales ubicados en cada uno de los bancos de capacitores. Con esta propuesta se plantea la alternativa de generar los comandos de operación en tiempo real para todos los bancos desde un centro de control, teniendo en cuenta el comportamiento global del sistema en todo momento. Adicionalmente, se pueden complementar las acciones de control locales de cada banco individual.

## 2. Formulación del problema

El problema consiste en determinar en qué barras del sistema de distribución deben ser adicionados condensadores (localización) y la cantidad de reactivos que deben inyectarse en ellas (tamaño), para reducir las pérdidas actuales del sistema, teniendo en cuenta que debe garantizarse un perfil de tensión adecuado. Se puede suponer que la red presenta un bajo nivel de desbalance y, por tanto, puede ser representado por un circuito equivalente monofásico. Así, el problema es descrito mediante el siguiente modelo matemático:

$$\begin{aligned} \min \quad & k_e \sum_{i=0}^{n_t} T_i p_i(x^i) + \sum_{k=1}^{n_c} f(u_k^o) + k_p \cdot p_0 \\ \text{sa.} \quad & G^i(x^i, u^i) = 0; i = 0, 1, \dots, n_t \\ & H^i(x^i) \leq 0; i = 0, 1, \dots, n_t \\ & 0 \leq u_k^i = u_k^o; k \in C_1 \\ & 0 \leq u_k^i \leq u_k^o; k \in C_2 \end{aligned} \quad (1)$$

En el modelo anterior se incluyen los siguientes parámetros:

- $n_t$ : Número de niveles de carga en que se discretiza la curva de duración de carga.
- $n_c$ : Número de barras candidatas para la localización de bancos de condensadores.
- $k_e$ : Costo de la energía en \$/kWh.
- $k_p$ : Costo de la potencia en \$/kW.
- $T_i$ : Duración del periodo de carga  $i$ .
- $p_i$ : Pérdidas en el sistema para el nivel de carga  $i$ .
- $p_o$ : Pérdidas en el sistema para el nivel de carga mas elevado.

$x^i$ : Variables de estado: Perfil de tensión en las barras del sistema de distribución.

$u^i$ : Variables de control: Potencia reactiva inyectada por los bancos de condensadores.

$H^i(x^i) \leq 0$ : Restricciones de operación para el nivel de carga  $i$ .

$G^i(x^i, u^i) = 0$ : Ecuaciones del flujo de carga.

$u_k^o$ : Tamaño del condensador instalado en la barra  $k$

$C_1$ : Conjunto de condensadores fijos.

$C_2$ : Conjunto de condensadores variables

$f(u_k^o)$ : Función discreta que describe los costos de instalación de bancos de condensadores.

En cada nivel de carga  $S_{nt}$  debe verificarse que la solución propuesta cumpla las restricciones del sistema, es decir, que no se excedan los límites de tensión y que la inversión sea justificada. Asimismo, la función objetivo considera un costo asociado a las pérdidas para cada nivel de carga, y otro adicional de penalización para el nivel más alto.

Este problema considera dos alternativas básicas: a) la ubicación de condensadores fijos: éstos se dimensionan teniendo en cuenta todos los niveles de carga, pero se presenta una sola propuesta para cada uno de ellos; b) la ubicación de condensadores variables: el dimensionamiento se hace para el nivel de carga más elevado, y luego se emplea un algoritmo adicional que determina la ubicación óptima y factible del *tap* del condensador en cada nivel de carga inferior. En este trabajo sólo se consideran los bancos de condensadores fijos y de un tamaño constante para todo el sistema. La figura 1 ilustra la curva de duración de carga discretizada en  $nt$  periodos; en dicha curva se representan los valores acumulados del consumo de energía de la carga, los cuales son medidos en un centro de carga.

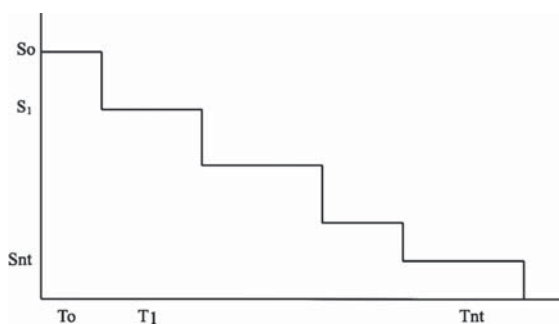


Figura 1. Curva de duración de carga discretizada.

En (1), la función objetivo se divide en dos partes. La primera representa los costos debidos a las pérdidas de energía;  $T_i$  es una fracción de  $T$  durante la cual el sistema opera al nivel de carga  $i$  con pérdidas  $p_i(x')$ , siendo  $k_e$  un parámetro que representa los costos de energía (\$US/KWh). La segunda parte representa los costos de instalación de los bancos. El problema es de difícil solución, porque el término  $f(u_k^0)$  de la función objetivo hace que ésta sea discreta no diferenciable; además, se presentan restricciones no lineales, existen variables discretas y continuas y presenta alta explosión combinatorial.

El problema se resuelve para dos tipos de operación: usando bancos fijos que operan de forma permanente en el sistema, o usando bancos variables operando con topología diferente para cada nivel de carga; este último es un tipo un problema mucho más complicado de resolver. Para la localización de bancos fijos, los condensadores son tratados como fuentes de potencia reactiva constantes, siendo válida la relación  $u^0 = u^1 = \dots = u^m$ .

El modelo matemático del problema de localización y tamaño óptimo de condensadores antes descrito puede ser resuelto a través de una metodología de optimización combinatorial. En este caso, el problema ha sido resuelto usando el algoritmo de optimización conocido como *tabu search* [2]. Así, la idea fundamental de este trabajo es utilizar una red neuronal entrenada con la información obtenida por el optimizador para comprobar las características de generalización de la red neuronal, en este tipo

de problemas, y agilizar el proceso de optimización, de tal forma que se obtengan buenos resultados en tiempo real al disminuir significativamente el esfuerzo computacional. Lo anterior hace posible que estas técnicas puedan ser usadas en sistemas automatizados.

### 3. Codificación de los parámetros de entrenamiento de la red neuronal

Para entrenar la red es necesario suministrar la información obtenida del optimizador (*tabu search*), que hace referencia a la potencia activa y reactiva inyectada en cada nodo, al número de condensadores y su ubicación para una condición de carga específica. Para cada nivel de la curva de duración de carga se debe entrenar la red considerando diferentes puntos de operación del sistema; mientras más grande sea el conjunto de parámetros de entrenamiento mejor será la generalización de la red al problema tratado. Sin embargo, el tiempo computacional invertido en el entrenamiento podría llegar a ser demasiado alto.

La red recibe en su entrada los valores de las curvas de carga para potencia activa y reactiva en cada nodo del sistema de distribución, y a su salida se le enseña el número de condensadores y su ubicación para una condición de carga específica, como se ilustra en la figura 2.

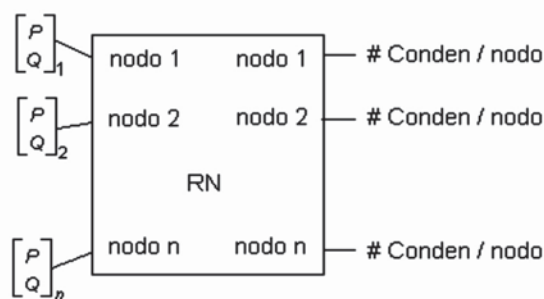


Figura 2. Parámetros de entrada y salida de la RNA.

Los datos de entrada de potencia activa y reactiva (números reales) son codificados de forma binaria [4] para asegurar que la red neuronal tenga un entrenamiento basado en valores discretos entre unos

y ceros y sea aplicada a cada nodo  $n$ , para cada nivel de carga  $S_m$  y cada tipo de carga. Por tanto, un sistema de  $n=9$  nodos con una curva de duración de carga discretizada en  $S_m=5$  niveles de carga, que además considera tres tipos de carga (residencial, comercial e industrial) debe realizar operaciones de codificación del orden de  $5^3 \times 9 = 1,125$ .

Para la codificación de los datos entregados por el optimizador con base en la aplicación del algoritmo de búsqueda tabú (ubicación y tamaño), primero se identifica el número máximo de condensadores instalados en el sistema de distribución ( $NC_{max}$ ) en todos los posibles casos evaluados. Después, con el valor de  $NC_{max}$ , para el nodo  $k$  se construye la representación binaria que se muestra en la figura 3; en ella, un «1» en la columna  $m$  indica que en el nodo  $k$  deben instalarse  $m$  bancos de condensadores de tamaño predefinido. Si en el vector todos los elementos son «0», significa que en el nodo  $k$  no deben instalarse condensadores; asimismo, debe tenerse en cuenta que sólo un elemento del vector puede valer «1»; el resto deben ser «0».

	1	2	...	m	...	$NC_{max}$
nodo <sub>k</sub>	0	0	...	1	...	0

Figura 3. Vector binario.

Al generalizar el arreglo anterior para cada uno de los nodos del sistema se obtiene, por ejemplo, el mostrado en la figura 4.

nodo <sub>1</sub>	0	1	...	0	...	0
nodo <sub>2</sub>	1	0	...	0	...	0
	⋮	⋮	⋱	⋮		⋮
nodo <sub>k</sub>	0	0	...	1	...	0
	⋮	⋮	⋱	⋮	⋱	⋮
nodo <sub>n</sub>	0	0	...	0	...	0

$n \times NC_{MAX}$

Figura 4. Codificación binaria usada para los parámetros de salida de la red.

El objetivo principal consiste en entrenar la red de tal manera que ésta entregue repuestas óptimas o muy cercanas al óptimo, para cualquier condición de demanda. Esto quiere decir que la red debe ser lo más general posible, para que sustituya el proceso que realiza el algoritmo de optimización. La diferencia en encontrar la respuesta a través de la aplicación de redes neuronales radica en la obtención de resultados en tiempo real.

#### 4. Estructura de la red neuronal utilizada

Entrenar una red neuronal artificial (RNA) es un proceso que consiste en modificar, por medio de algoritmos, el valor de los pesos y *bias* asociados a cada neurona [4], con el fin de que a partir de unos datos presentados en la entrada la red pueda generar una salida acorde con los datos de entrenamiento usados.

La forma de decidir cuándo está entrenada la red depende del tipo usado y de la salida deseada; en el caso del aprendizaje supervisado para una RNA [5] se tiene un conjunto de datos correlacionales (patrones de entrenamiento) pero no se conoce la función o relación matemática que los representa; al propagar hacia delante cada uno de estos patrones se obtiene una respuesta en la salida de la RNA, la cual se compara con la salida deseada permitiendo obtener el error del desempeño de la red. El error es una función matemática que depende del valor actual de todos los pesos y *bias* de las neuronas; por tanto, la modificación de estos últimos busca reducir su valor.

Generalmente, una RNA posee varias capas y en cada una se tiene más de una neurona; esto define la cantidad de pesos y *bias* de la red. Aún para redes muy pequeñas, su comportamiento es altamente no lineal. Las redes neuronales artificiales buscan simular las tres características básicas de una red neuronal biológica [6]: a) el *procesamiento paralelo*: la red tiene la capacidad de incorporar y analizar simultáneamente muchos datos; b) la *memoria distribuida*: cada neurona tiene parte de la información necesaria para obtener una respuesta



en la salida; c) la *adaptabilidad*: consiste en explotar la habilidad de las neuronas de cambiar sus pesos dependiendo de la estimulación y la respuesta deseada. Las anteriores características permiten a estas redes, a partir de casos particulares, aprender y generalizar las relaciones entre un conjunto de datos en la entrada con un conjunto de datos en la salida, incluso cuando no se conoce la función matemática que los representa.

Existen muchas definiciones para las RNA, pero, en general, éstas pueden representarse como se muestra en la figura 4. Haykin [4] establece que una RNA es un grafo dirigido que cumple con las siguientes propiedades: a) a cada nodo  $i$  se asocia una variable de estado  $y_i$ ; b) a cada conexión  $(i,j)$  de los nodos  $i$  y  $j$  se asocia un peso sináptico  $w_{ji}$  y  $w_{ji} \in R$ ; c) a cada nodo  $j$  se asocia un *bias*  $b_j$  y  $b_j \in R$ ; d) para cada nodo  $j$  se define una función de transferencia  $f_j(y_j, w_{ji}, b_j)$  que depende de los pesos, conexiones, *bias* y estados de los nodos  $i$  conectados a él; esta función determina el nuevo estado del nodo.

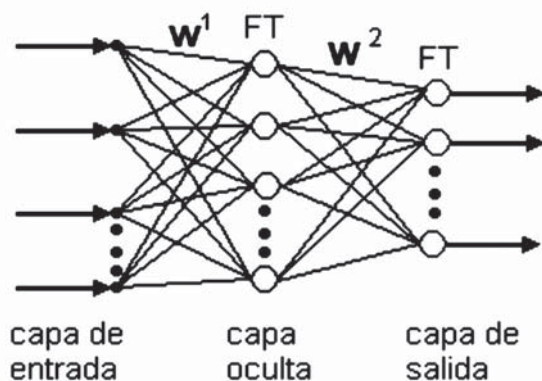


Figura 5. Esquema de red neuronal multicapa.

En la figura 5,  $w^1$  y  $w^2$  son las matrices de pesos entre capas y  $FT$  es la matriz de la función de transferencia de cada capa. La figura 6 muestra la estructura de una neurona artificial que pertenece a la capa de salida de una RNA multicapa.

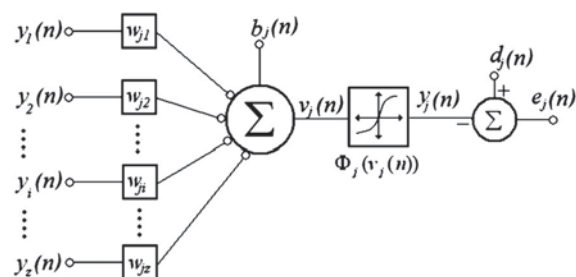


Figura 6. Modelo de una neurona en una RNA.

En la figura 6:

$n$  denota una iteración cuando se propaga en la red el  $n$ -ésimo patrón del conjunto de entrenamiento.

$y_j(n)$  es la salida de la neurona  $j$ .

$w_{ji}(n)$  es el peso de conexión de la entrada  $i$  en la neurona  $j$ .

$b_j(n)$  es el parámetro *bias* de la neurona  $j$ .

$v_j(n)$  es la regla de propagación que determina el potencial de activación de la neurona  $j$ .

$f_j$  es la función de transferencia de la neurona  $j$ .

$d_j(n)$  es la salida esperada de la neurona  $j$  al propagar el patrón  $n$ .

$e_j(n)$  corresponde al error en la salida de la neurona  $j$  al propagar el  $n$ -ésimo patrón.

$N$  es el número de patrones de entrenamiento.

Un algoritmo de entrenamiento de una RNA modifica el valor de los pesos de acuerdo con la siguiente expresión:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) \quad (2)$$

En (2),  $\Delta w_{ji}$  es el valor que se obtiene por medio de una regla que define el algoritmo de entrenamiento. Para el error de la red cuando se propaga el  $n$ -ésimo patrón de entrenamiento  $E(n)$  se tienen dos posibilidades de medir el desempeño:

$$E(n) = \frac{1}{2N} \sum_z \sum_j (e_j(n)^z)^2 \quad (3)$$

$$E(n) = \frac{1}{2} \sum_j e_j(n)^2 \quad (4)$$

La expresión (3) define el error global en la RNA, sobre todo el conjunto de aprendizaje o sobre los  $N$  patrones, mientras que la (4) define el error medio cuadrático (MSE) instantáneo o de un patrón y es una aproximación a (3), con la ventaja de requerir un menor esfuerzo computacional; la segunda expresión se utiliza en la implementación realizada en este trabajo.

El algoritmo de propagación hacia atrás (BP) modifica  $w_{ji}(n)$  respecto a  $E(n)$  usando la siguiente expresión:

$$\Delta w_{ji}(n) = \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (5)$$

Utilizando (5), el algoritmo de propagación hacia atrás establece que:

$$\Delta w_{ji}(n) = -\delta_j(n) y_i(n) \quad (6)$$

En (6),  $d_j(n)$  es el gradiente local que mide el error que introduce una neurona o una capa de la red en el error global. Si  $j$  es una neurona de la capa de salida se tiene:

$$\delta_j(n) = e_j(n) \Phi_j'(v_j(n)) \quad (7)$$

Si la neurona  $j$  pertenece a una capa oculta, se tiene:

$$\delta_j(n) = \Phi_j'(v_j(n)) \sum_k \delta_k(n) w_{kj}(n) \quad (8)$$

Por lo anterior, el algoritmo *BP* modifica iterativamente los pesos con las expresiones (6) y (7) hasta que (3) ó (4) alcancen un valor de tolerancia definido, lo cual significa que la red ha aprendido el conjunto de entrenamiento. De acuerdo con la naturaleza del problema puede determinarse con facilidad la cantidad de neuronas en las capas de entrada y salida; sin embargo, el número de capas ocultas y la cantidad de neuronas en ellas no sigue ningún

patrón o regla definida, por lo que se acude más a la experiencia en la solución de problemas similares.

## 5. Sistema de prueba y resultados

Se eligió el sistema de 13 barras que se muestra en la figura 7; sus datos son mostrados en la tabla 1.

Datos de barra				
Barra	Generación	Demanda	Voltaje	Tipo
1	0+0i	0.000+0.000i	1	Slack
2	0+0i	0.184+0.046i	1	Carga
3	0+0i	0.098+0.034i	1	Carga
4	0+0i	0.179+0.045i	1	Carga
5	0+0i	0.160+0.184i	1	Carga
6	0+0i	0.161+0.060i	1	Carga
7	0+0i	0.078+0.011i	1	Carga
8	0+0i	0.115+0.006i	1	Carga
9	0+0i	0.098+0.013i	1	Carga
10	0+0i	0.164+0.020i	1	Carga
11	0+0i	0.161+0.060i	1	Carga
12	0+0i	0.115+0.006i	1	Carga
13	0+0i	0.098+0.034i	1	Carga

Tabla 1. Datos del sistema.

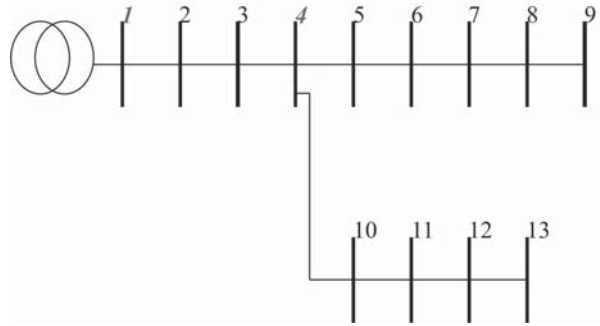
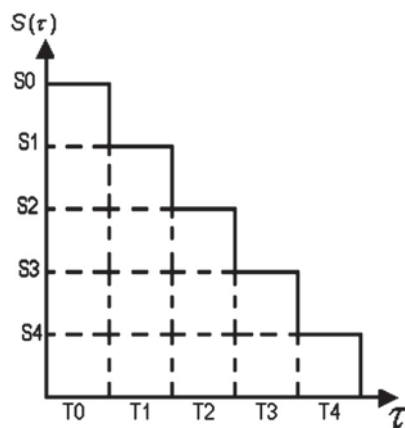


Figura 7. Sistema de prueba

El sistema mostrado se adaptó para los siguientes parámetros:  $K_e = \$0,06/kWh$ ,  $K_p = \$168/kW/año$ ,  $K_c = \$4,9/Kvar$ . Para la curva de duración discretizada se tienen los siguientes niveles de carga normalizados:  $S_0 = 1,0$ ,  $S_1 = 0,8$ ,  $S_2 = 0,6$ ,  $S_3 = 0,4$ ,  $S_4 = 0,2$ , y las duraciones de cada nivel durante un año se distribuyen de la siguiente manera:  $T_0 = 1.460$  horas,  $T_1 = 1.825$  horas,  $T_2 = 1.095$  horas,  $T_3 = 1.852$  horas,  $T_4 = 2.555$  horas (figura 8).



**Figura 8.** Curva de duración discretizada en cinco niveles de carga.

Del análisis de los resultados arrojados por el optimizador empleando el algoritmo de búsqueda tabú se concluye que  $NC_{\max} = 5$ , es decir, se permiten hasta cinco bancos de condensadores de 300 kVar en cada barra. Se considera un horizonte de tiempo de un año. Para una operación sin compensación, la función objetivo representa un costo de \$329.039 por pérdidas de energía. Se utiliza una red multica- pa conformada por una capa de entrada, una oculta y una de salida, cuyas características se muestran en la tabla 2.

	Capa de entrada	Capa oculta	Capa de salida
Función de transferencia	Tangente sigmoideal (Ecuación 9)	Tangente sigmoideal (Ecuación 9)	Logaritmo sigmoideal (Ecuación 10)
Número de neuronas	2*(# nodos -1)	Variable	(# condensadores)* (# nodos -1)

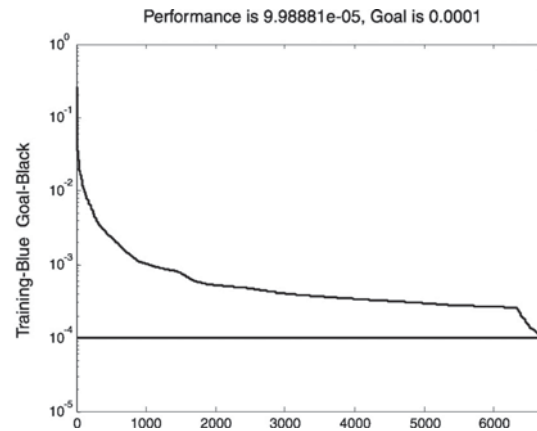
**Tabla 2.** Características de la red neuronal.

Las expresiones utilizadas que definen la función de transferencia para las capas de entrada y oculta y para la capa de salida, respectivamente, son:

$$\tan(\text{sig}(n)) = \frac{2}{1 + e^{-2n}} \quad (9)$$

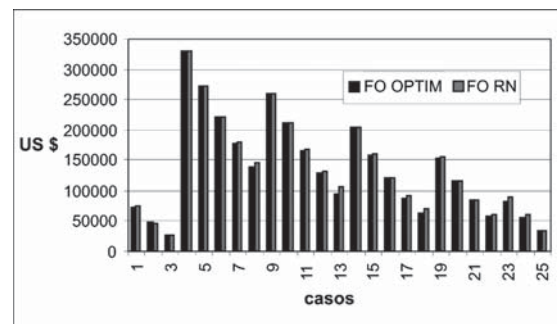
$$\log(\text{sig}(n)) = \frac{1}{1 + e^{-n}} \quad (10)$$

Se consideraron tres tipos de carga (comercial, industrial y residencial) y se usa la misma curva de duración para los tres. El 80% de los resultados obtenidos para todos los casos posibles se toman para entrenar la red neuronal, y el 20% restante como conjunto de validación. La gráfica 1 ilustra el proceso de aprendizaje sufrido por la red neuronal propuesta.



**Gráfica 1.** Convergencia de la red.

La gráfica 1 permite concluir que la red neuronal en verdad aprendió a optimizar un sistema de 13 barras cuya función objetivo se plantea en función de las pérdidas y la inversión. Este resultado cobra mayor importancia si se considera que el problema tratado es de programación no lineal entero mixto (PNLIM). Los resultados comparativos se ilustran en la gráfica 2; allí se aprecia que en todos los casos el optimizador neuronal alcanza o se aproxima significativamente al optimizador heurístico basado en el método de búsqueda tabú.



**Gráfica 2.** Comparación de los valores de las funciones objetivo.



En las tablas 3 y 4 se muestran las soluciones encontradas por el algoritmo de búsqueda tabú y la red neuronal, respectivamente. Esta última sugiere colocar dos bancos de condensadores de 300 kVar cada uno (600 kVar) en la barra 4, y un único banco de 300 kVar en la barra 6. El algoritmo de búsqueda tabú sugiere colocar un único banco en la barra 6.

**Tabla 3.** Resultados del algoritmo de búsqueda tabú de la red neuronal.

Barra	Condensador (kVar)
6	300

**Tabla 4.** Resultados

Barra	Condensador (kVar)
4	600
6	300

El resultado obtenido al agregar la compensación de reactivos al sistema usando el algoritmo basado en búsqueda tabú produjo una recuperación de pérdidas de \$46.264, mientras que empleando la red neuronal la recuperación fue de \$47.511. Este resultado se encontró usando 65 neuronas en la capa oculta de la red (definidas con un procedimiento iterativo de ajuste y error hasta encontrar una buena solución) y empleando tiempos computacionales significativamente menores en la obtención del resultado, por parte de la RNA (aproximadamente 20 veces más rápido). El proceso de entrenamiento de la red se realizó en un computador PC con procesador de 700 MHz y tardó tres horas, aproximadamente.

## 6. Conclusiones y recomendaciones

La discretización de la curva de duración define el grado de precisión del modelo matemático. Mientras mayor sea el número de niveles de carga más

aproximadas serán las respuestas al comportamiento real del sistema. Sin embargo, el esfuerzo computacional aumenta exponencialmente y el tiempo de entrenamiento de la red puede llegar a ser inviable.

Para los datos de entrada y de salida de la red, la codificación del sistema resulta ser un proceso de extrema importancia que define la calidad de las respuestas y los esfuerzos computacionales de los procesos de entrenamiento y optimización de la red. Por tanto, es importante verificar los resultados obtenidos con otros tipos de codificación y con diferentes características para la RNA.

Las técnicas propuestas tienen importancia en el estudio de sistemas de gran tamaño, en los que es necesario obtener soluciones en tiempo real y en forma periódica. Así, ellas pueden ser implementadas en otros tipos de problemas asociados, por ejemplo: en el despacho eléctrico, en el cual las condiciones de altas velocidades computacionales en la obtención de la respuesta son necesarias, los sistemas son de gran tamaño y los procesos de optimización son ejecutados con alta periodicidad.

El problema de ubicación y tamaño óptimo de condensadores en sistemas de distribución abordado en este trabajo sólo considera los de tipo fijo. Para la inclusión de condensadores variables, la técnica basada en RNA sólo cambia en la codificación de los parámetros entregados por el optimizador.

La implementación de las redes neuronales en este tipo de problemas de optimización combinatoria de difícil solución permite aprovechar las características de generalización de la red para obtener respuestas óptimas o subóptimas de buena calidad. Además, la optimización se realiza sobre todo el sistema desde un centro de control, lo cual abre la posibilidad de disminuir los costos del sistema físico de control individual de cada banco.

## Referencias bibliográficas

- [1] Gallego R., Escobar A. y Rodas D. (2002). *Algoritmo genético especializado para la localización óptima de condensadores en sistemas de distribución de energía eléctrica*, I Encuentro Andino y II Congreso Colombiano de Investigación de Operaciones, Bogotá.
- [2] Gallego R., Monticelli A. and Romero R. (2001). *Optimal Condensador Placement in Radial Distribution Networks*, IEEE Transactions on Power Systems, Vol. 16, núm. 4.
- [3] Gallego R.A., Romero R.A., Escobar A., (1999). *Algoritmos genéticos*, 1.<sup>a</sup> ed., Maestría en Ingeniería Eléctrica, Universidad Tecnológica de Pereira.
- [4] Haykin, S. (1999). *Neural Network. A Comprehensive Foundation*. 2nd ed. Prentice-Hall.
- [5] Mollen M.F. (1993). *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*, Neural Networks, 6(3), pp. 525-533.
- [6] Sanz, A., Martín del Ríos, B. (2002). *Redes neuronales y sistemas difusos*. Ed. Alfaomega, 2<sup>a</sup> ed.
- [7] Grainger, J.J, Lee, S. H. (1982). *Capacity Release by Shunt Condensador Placement on Distribution Feeders: A New Voltage-Dependency Model*, IEEE Transactions on Power Apparatus and Systems, Vol. PAS-101, núm. 5.
- [8] R Romero, R A Gallego, and A Monticelli, (1996). *Transmission System Expanssion Planning by Simulated Annealing*, Trans on Power System, Vol 11, núm. 1, pp 364-369.
- [9] Gallego R. A., Romero R. and Salazar H. *Artificial Neural Network and Clustering Techniques Applied in the Reconfiguration of Distribution Systems*. IEEE Transaction on Power Delivery. En prensa.